The JHU Machine Translation Systems for WMT 2017

The JHU Machine Translation Systems for WMT 2017

Shuoyang Ding[†] **Huda Khayrallah**[†] **Philipp Koehn**[†]
Matt Post[‡] **Gaurav Kumar**[†] **and Kevin Duh**[‡]
[†]Center for Language and Speech Processing
[‡]Human Language Technology Center of Excellence
Johns Hopkins University
{dings, huda, phi}@jhu.edu,
{post, gkumar, kevinduh}@cs.jhu.edu

Abstract

This paper describes the Johns Hopkins University submissions to the shared translation task of EMNLP 2017 Second Conference on Machine Translation (WMT 2017). We set up phrase-based, syntax-based and/or neural machine translation systems for all 14 language pairs of this year's evaluation campaign. We also performed neural rescoring of phrase-based systems for English-Turkish and English-Finnish.

1 Introduction

The JHU 2017 WMT submission consists of phrase-based systems, syntax-based systems and neural machine translation systems. In this paper we discuss features that we integrated into our system submissions. We also discuss lattice rescoring as a form of system combination of phrase-based and neural machine translation systems.

The JHU phrase-based translation systems for our participation in the WMT 2017 shared translation task are based on the open source Moses toolkit (Koehn et al., 2007) and strong baselines of our submission last year (Ding et al., 2016). The JHU neural machine translation systems were built with the Nematus (Sennrich et al., 2016c) and Marian (Junczys-Dowmunt et al., 2016) toolkits. Our lattice rescoring experiments are also based on a combination of these three toolkits.

2 Phrase-Based Model Baselines

Although the focus of research in machine translation has firmly moved onto neural machine translation, we still built traditional phrase-based statistical machine translation systems for all language pairs. These submissions also serve as a baseline of where neural machine translation systems stand with respect to the prior state of the art.

Our systems are very simmilar to the JHU systems from last year (Ding et al., 2016).

2.1 Configuration

We trained our systems with the following settings: a maximum sentence length of 80, grow-diag-final-and symmetrization of GIZA++ alignments, an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield, 2011) used at runtime, hierarchical lexicalized reordering (Galley and Manning, 2008), a lexically-driven 5-gram operation sequence model (OSM) (Durrani et al., 2013) with 4 count-based supportive features, sparse domain indicator, phrase length, and count bin features (Blunsom and Osborne, 2008; Chiang et al., 2009), a distortion limit of 6, maximum phrase-length of 5, 100-best translation options, compact phrase table (Junczys-Dowmunt, 2012) minimum Bayes risk decoding (Kumar and Byrne, 2004), cube pruning (Huang and Chiang, 2007), with a stack-size of 1000 during tuning and 5000 during test and the no-reordering-over-punctuation heuristic (Koehn and Haddow, 2009). We optimize feature function weights with k-best MIRA (Cherry and Foster, 2012).

We used POS and morphological tags as additional factors in phrase translation models (Koehn and Hoang, 2007) for the German-English language pairs. We also trained target sequence models on the in-domain subset of the parallel corpus using Kneser-Ney smoothed 7-gram models. We used syntactic preordering (Collins et al., 2005) and compound splitting (Koehn and Knight, 2003) for the German-to-English systems. We did no language-specific processing for other languages.

We included Och cluster language model, with 4 additional language models trained on 50, 200,

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 276–282
Copenhagen, Denmark, September 711, 2017.

Language Pair	Sentences
German–English	21,243
Czech–English	21,730
Finnish–English	2,870
Latvian–English	984
Russian-English	11,824
Turkish–English	1,001
Chinese–English	1,000

Table 1: Tuning set sizes for phrase and syntax-based system

500, and 2000 clusters (Och, 1999) using `mkcls`. In addition, we included a large language model based on the CommonCrawl monolingual data (Buck et al., 2014).

The systems were tuned on a very large tuning set consisting of the test sets from 2008-2015, with a total of up to 21,730 sentences (see Table 1). We used `newstest2016` as development test set. Significantly less tuning data was available for Finnish, Latvian, and Turkish.

2.2 Results

Table 2 shows results for all language pairs, except for Chinese–English, for which we did not built phrase-based systems. Our phrase-based systems were clearly outperformed by NMT systems for all language pairs, by a difference of 3.2 to 8.3 BLEU points. The difference is most dramatic for languages with rich morphology (Turkish, Finnish).

3 Syntax-based Model Baselines

We built syntax-based model baselines for both directions of Chinese-English language pairs because our previous experiments indicate that syntax-based machine translation systems generally outperform phrase-based machine translation systems by a large margin. Our system setup was largely based on our syntax-based system setup for last year's evaluation (Ding et al., 2016).

3.1 Configuration

Our syntax-based systems were trained with all the CWMT and UN parallel data provided for the evaluation campaign. We also used the monolingual data from news crawl 2007-2016, the English Gigaword, and the English side of Europarl corpus. The CWMT 2008 multi-reference dataset were used for tuning (see statistics in Table 1).

For English data, we used the scripts from Moses (Koehn et al., 2007) to tokenize our data, while for Chinese data we carried out word segmentation with Stanford word segmenter (Chang et al., 2008). We also normalized all the Chinese punctuations to their English counterparts to avoid disagreement across sentences. We parsed the tokenized data with Berkeley Parser (Petrov and Klein, 2007) using the pre-trained grammar provided with the toolkit, followed by right binarization of the parse. Finally, truecasing was performed on all the English texts. Due to the lack of casing system, we did not perform truecasing for any Chinese texts.

We performed word alignment with fast-align (Dyer et al., 2013) due to the huge scale of this year's training data and grow-diag-final-and heuristic for alignment symmetrization. We used the GHKM rule extractor implemented in Moses to extract SCFG rules from the parallel corpus. We set the maximum number of nodes (except target words) in the rules (`MaxNodes`) to 30, maximum rule depth (`MaxRuleDepth`) to 7, and the number of non-part-of-speech, non-leaf constituent labels (`MaxRuleSize`) to 7. We also used count bin features for the rule scoring as our phrase-based systems (Blunsom and Osborne, 2008)(Chiang et al., 2009). We used the same language model and tuning settings as the phrase-based systems.

While BLEU score was used both for tuning and our development experiments, it is ambiguous when applied for Chinese outputs because Chinese does not have explicit word boundaries. For discriminative training and development tests, we evaluate the Chinese output against the automatically-segmented Chinese reference with `multi-bleu.perl` scripts in Moses (Koehn et al., 2007).

3.2 Results

Our development results on `newsdev2017` are shown in Table 3. Similar to the phrase-based system, the syntax-based system is also outperformed by NMT systems for both translation directions.

4 Neural Machine Translation[1]

We built and submitted neural machine translation systems for both Chinese-English and English-Chinese language pairs. These systems are trained

[1] All the scripts and configurations that were used to train our neural machine translation systems can be retrieved at `https://github.com/shuoyangd/nmt4clsp`

Language Pair	JHU 2016	Baseline	Och LM	Och+CC LM	Och+CC LM	Best NMT
	newstest2016				newstest2017	
English-Turkish	9.22	9.22	9.11	9.30	9.8	18.1 +8.3
Turkish-English	12.94	13.03	12.92	12.83	12.6	20.1 +7.5
English-Finnish	13.76	14.12	14.04	13.99	14.5	20.7 +6.2
Finnish-English	19.08	19.72	19.36	19.16	20.5	-
English-Latvian	-	18.66	18.71	18.85	14.4	20.1 +5.7
Latvian-English	-	25.82	26.03	26.12	16.8	20.0 +3.2
English-Russian	23.99	21.45		23.16	25.3	29.8 +4.5
Russian-English	27.88	24.47		27.22	31.5	34.7 +3.2
English-Czech	23.56			23.05	19.1	22.8 +3.7
Czech-English	30.37	29.84	29.98	29.80	26.5	30.9 +4.4
English-German	28.35	28.95		28.39	21.6	28.3 +6.7
German-English	34.50		34.20	33.87	29.7	35.1 +5.4

Table 2: Phrase-Based Systems (cased BLEU scores)

with all the CWMT and UN parallel data provided for the evaluation campaign and `newsdev2017` as the development set. For the back-translation experiments, we also included some monolingual data from new crawl 2016, which is back-translated with our basic neural machine translation system.

4.1 Preprocessing

We started by following the same preprocessing procedures for our syntax-based model baselines except that we didn't do parsing for our training data for neural machine translation systems. After these procedures, we then applied Byte Pair Encoding (BPE) (Sennrich et al., 2016c) to reduce the vocabulary size in the training data. We set the number of BPE merging operations as 49500. The resulting vocabulary size for Chinese and English training data are 64126 and 35335, respectively.

4.2 Training

We trained our basic neural machine translation systems (labeled `base` in Table 3) with Nematus (Sennrich et al., 2017). We used batch size 80, vocabulary size of 50k, word dimension 500 and hidden dimension 1024. We performed dropout with dropout rate 0.2 for the input bi-directional encoding and the hidden layer, and 0.1 for the source and target word embedding. To avoid gradient explosion, gradient clipping constant 1.0 was used. We chose AdaDelta (Zeiler, 2012) as the optimization algorithm for training and used decay rate $\rho = 0.95$, $\varepsilon = 10^{-6}$.

We performed early stopping according to the validation error on the development set. The validation were carried out every 5000 batch updates. The early stopping was triggered if the validation error does not decrease for more than 10 validation runs, i.e. more than 50k batch updates.

4.3 Decoding and Postprocessing

To enable faster decoding for validation, test and back-translation experiments (in Section 4.4), we used the decoder from Marian (Junczys-Dowmunt et al., 2016) toolkit. For all the steps where decoding is involved, we set the beam size of RNN search to 12.

The postprocessing we performed for the final submission starts with merging BPE subwords and detokenization. We then performed de-trucasing for English output, while for Chinese output we re-normalized all the punctuations to their Chinese counterparts. Note that for fair comparison, we used the same evaluation methods for English-Chinese experiments as we did for the English-Chinese syntax-based system, which means we do not detokenize our Chinese output for our development results.

4.4 Enhancements: Back-translation, Right-to-left models, Ensembles

To investigate the effectiveness of incorporating monolingual information with back-translation (Sennrich et al., 2016b), we continued training on top of the `base` system to build another system (labeled `back-trans` below) that has some exposure to the monolingual data. Due to the time and hardware constraints, we only took a random sample of

Language Pairs	Syntax	base single	base ensemble	back-trans single	back-trans ensemble
Chinese-English	16.22	17.81	**18.46**	17.52	18.16
English-Chinese	14.43	17.22	17.95	17.76	**18.60**

Table 3: Chinese-English and English-Chinese System Development Results on newsdev2017 (cased BLEU scores). Bold scores indicate best and submitted systems.

2 million sentences from news crawl 2016 monolingual corpus and 1.5 million sentences from preprocessed CWMT Chinese monolingual corpus from our syntax-based system run and back-translated them with our trained base system. These back-translated pseudo-parallel data were then mixed with an equal amount of random samples from real parallel training data and used as the data for continued training. All the hyperparameters used for the continued training are exactly the same as those in the initial training stage.

Following the effort of (Liu et al., 2016) and (Sennrich et al., 2016a), we also trained right-to-left (r2l) models with a random sample of 4 million sentence pairs for both translation directions of Chinese-English language pairs, in the hope that they could lead to better reordering on the target side. But they were not included in the final submission because they turned out to hurt the performance on development set. We conjecture that our r2l model is too weak compared to both base and back-trans models to yield good reordering hypotheses.

We performed model averaging over the 4-best models for both base and back-trans systems as our combined system. The 4-best models are selected among the model dumps performed every 10k batch updates in training, and we select the models that has the highest BLEU scores on the development set. The model averaging was performed with the average.py script in Marian (Junczys-Dowmunt et al., 2016).

4.5 Results

Results of our neural machine translation systems on newsdev2017 are also shown in Table 3. Both of our neural machine translation systems outputperform their syntax-based counterparts by 2-4 BLEU points.

The results also indicate that the 4-best averaging ensemble uniformly performs better than single systems. However, the back-translation experiments for Chinese-English system do not improve performance. We hypothesize that the amount of our back-translated data is not sufficient to improve the model. Experiments with full-scale back-translated monolingual data are left for future work.

5 Rescoring

We use neural machine translation (NMT) systems to rescore the output of the phrase-based machine translation (PBMT) systems. We use two methods to do this, 500-best list rescoring, and lattice rescoring. Rescoring was performed on English-Turkish, and English-Finnish translation tasks. We combined the baseline PBMT models from Table 2, with basic NMT systems.

5.1 NMT Systems

We build basic NMT systems for this task. We preprocess the data by tokenizing, truecasing, and applying Byte Pair Encoding (Sennrich et al., 2015) with 49990 merge operations. We trained the NMT systems with Nematus (Sennrich et al., 2017) on the released training corpora. We used the following settings: batch size of 80, vocabulary size of 50000, word dimension 500, and hidden dimension 1000. We performed dropout with a rate of 0.2 for the input bi-directional encoding and the hidden layer, and 0.1 for the source and target word embedding. We used Adam as the optimizer (Kingma and Ba, 2014).

We performed early stopping according to the validation error on the development set. Validation was carried out every 20000 batch updates. The early stopping was triggered if the validation error does not decrease for more than 10 validation runs, if early stopping is not triggered, we run for a maximum of 50 epochs.

We create ensembles by averaging the 3 best validation models with the average.py script in Marian (Junczys-Dowmunt et al., 2016).

Language Pair	PBMT	NMT	NMT-Ens	N-best	Lattice	N-best	Lattice
	newstest2016					newstest2017	
English-Turkish	9.2	8.1	8.5	9.4	**9.9**	9.4	**10.4**
English-Finnish	14.1	12.6	13.6	14.6	**15.5**	14.3	**16.0**

Table 4: Comparison of PBMT, NMT, NMT-Ensembles, and neural rescoring of PBMT output in the form of N-best lists or lattices (cased BLEU scores)

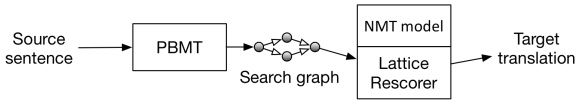

Figure 1: The neural lattice rescorer pipeline.

5.2 500-best Rescoring

We rescore 500-best candiate lists by first generating 500-best lists from Moses (Koehn et al., 2007) using the `-N-best-list` flag. We then use the Nematus (Sennrich et al., 2017) N-best list rescoring to rescore the list using our NMT model.

5.3 Lattice Rescoring

We also rescore PBMT lattices. We generate search graphs from the PBMT system by passing the `-output-search-graph` parameter to Moses. The search graphs are then converted to the OpenFST format (Allauzen et al., 2007) and operations to remove epsilon arcs, determinize, minimize and topsort are applied. Since the search graphs may be prohibitively large in size, we prune them to a threshold; we tune this threshold.[2]

The core difficulty in lattice rescoring with NMT is that its RNN architecture does not permit efficient recombination of hypotheses on the lattice. Therefore, we apply a stack decoding algorithm (similar to the one used in PBMT) which groups hypotheses by the number of target words (the paper describing this work is under review). Figure 5.3 describes this pipeline.

5.4 Results

We use `newstest2016` as a developement set, and report the official results from `newstest2017`.

Tables 5 and 6 show the development set results for pruning thresholds of .1, .25, and .5 and stack sizes of 1, 10, 100, 1000. We chose not to use a stack size of 1000 in our final systems because the improvement in devset BLEU over a stack size of 100 is not large. For our final English-Turkish system, we use a pruning threshold of .25 and a stack size of 100; for our final English-Finnish system we use a pruning threshold of .5 and a stack size of 100.

[2]Pruning removes arcs that do not appear on a lattice path whose score is within than $t \otimes w$, where w is the weight of the FSTs shortest path, and t is the pruning threshold.

	.1	.25	.5
1	9.60	9.51	9.11
10	9.82	9.86	9.28
100	9.86	9.90	9.43
1000	9.88	**9.92**	-

Table 5: Grid search on the pruning (.1, .25, .5) and stack parameters (1, 10, 100, 1000) for English-Turkish `newstest2016` (cased BLEU)

	.1	.25	.5
1	14.85	15.06	14.96
10	14.92	15.30	15.32
100	14.92	15.33	15.49
1000	14.94	15.29	**15.53**

Table 6: Grid search on the pruning (.1, .25, .5) and stack parameters (1, 10, 100, 1000) for English-Finnish `newstest2016` (cased BLEU)

Table 4 shows development results for the baseline PBMT, NMT systems, as well as the NMT ensembles, 500-best rescoring, and lattice rescoring. We also report test results for the 500-best rescoring, and lattice rescoring. On `newstest2016`, lattice rescoring outperforms 500-best rescoring by .5-1.1 BLEU, and on `newstest2017`, lattice rescoring outperforms 500-best rescoring by 1-1.7 BLEU. 500-best rescoring also outperforms PBMT, NMT system, and the NMT ensembles. While these results are not competitive with the best systems on `newstest2017` in the evaluation campaign, it is interesting to note that lattice rescoring gave good performance among the models we compared. For future work it is worth re-running the lattice rescoring experiment using stronger baseline PBMT and NMT models.

6 Conclusion

We submitted phrase-based systems for all 14 language pairs, syntax-based systems for 2 pairs, neural systems for 2 pairs, and two types of rescored systems for 2 pairs. While many of these systems underperformed neural systems, they provide a strong baseline to compare the new neural systems to the previous state-of-the-art phrase-based systems. The gap between our neural systems and the top performing ones can be partially explained by a lack of large-scale back-translated data, which we plan to include in future work.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*. Springer, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. `http://www.openfst.org`.

Phil Blunsom and Miles Osborne. 2008. Probabilistic Inference for Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, pages 215–223.

Christian Buck, Kenneth Heafield, and Bas Van Ooyen. 2014. N-gram Counts and Language Models from the Common Crawl. *LREC* 2:4.

Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In *Proceedings of the third workshop on statistical machine translation*. Association for Computational Linguistics, pages 224–232.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Montréal, Canada, pages 427–436.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 New Features for Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Boulder, Colorado, pages 218–226.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 531–540.

Shuoyang Ding, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. 2016. The JHU Machine Translation Systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 272–280.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sofia, Bulgaria.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii, pages 848–856.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, United Kingdom, pages 187–197.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 144–151.

M. Junczys-Dowmunt. 2012. A phrase table without phrases: Rank encoding for better phrase table compression. In Mauro Cettolo, Marcello Federico, Lucia Specia, and Andy Way, editors, *Proceedings of th 16th International Conference of the European Association for Machine Translation (EAMT)*. pages 245–252.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*. Seattle, WA.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980.

Philipp Koehn and Barry Haddow. 2009. Edinburgh's Submission to all Tracks of the WMT 2009 Shared Task with Reordering and Speed Improvements to Moses. In *Proceedings of the Fourth Workshop*

on Statistical Machine Translation. Association for Computational Linguistics, Athens, Greece, pages 160–164.

Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 868–876.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL*. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2003. "empirical methods for compound splitting". In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

Shankar Kumar and William J. Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *HLT-NAACL*. pages 169–176.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of NAACL-HLT*. pages 411–416.

Franz Josef Och. 1999. An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pages 71–76.

Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *HLT-NAACL*. volume 7, pages 404–411.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 65–68.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. *CoRR* abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*. pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .